

Debugging Parallel Programs



Rebecca Hartman-Baker
hartmanbakrj@ornl.gov



U.S. DEPARTMENT OF
ENERGY



OAK RIDGE NATIONAL LABORATORY

MANAGED BY UT-BATTELLE FOR THE DEPARTMENT OF ENERGY

Outline

- Introduction to debugging parallel programs
- Methods of debugging parallel programs
- Why use a debugger?
- What can a debugger do for me?

Debugging Parallel Programs

- Parallel programs are hard to debug
 - Serial programs are hard enough!
 - Parallel programming adds complexity
 - Must consider concurrency, synchronization, communication, blocking/non-blocking calls, etc.
- Ways to debug parallel programs
 - Print statement debugging
 - Code reading and role-playing (I'm P0, you're P1)
 - Arts & Crafts/drawing
 - *Using a debugger*

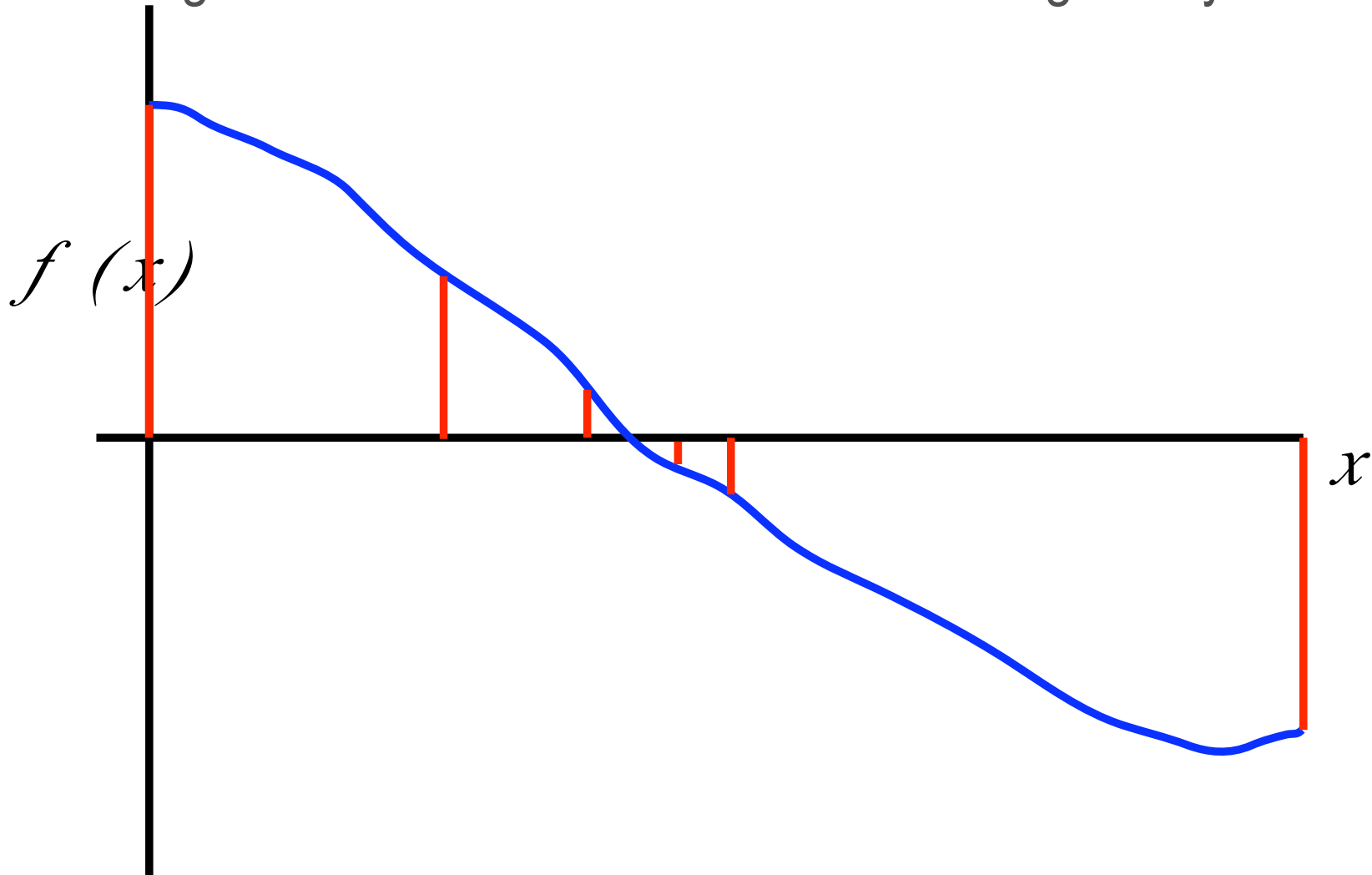
Print-Statement Debugging

- Each processor dumps print statements to `stdout` or into individual output files, e.g., `log.0001`, `log.0002`, etc.
- *Advantages:* easy to implement, independent of platform or available resources
- *Disadvantages:* time-consuming, extraneous information in log files, tedious, not scalable (imagine 100K “Hi from processor x” messages?!?!)



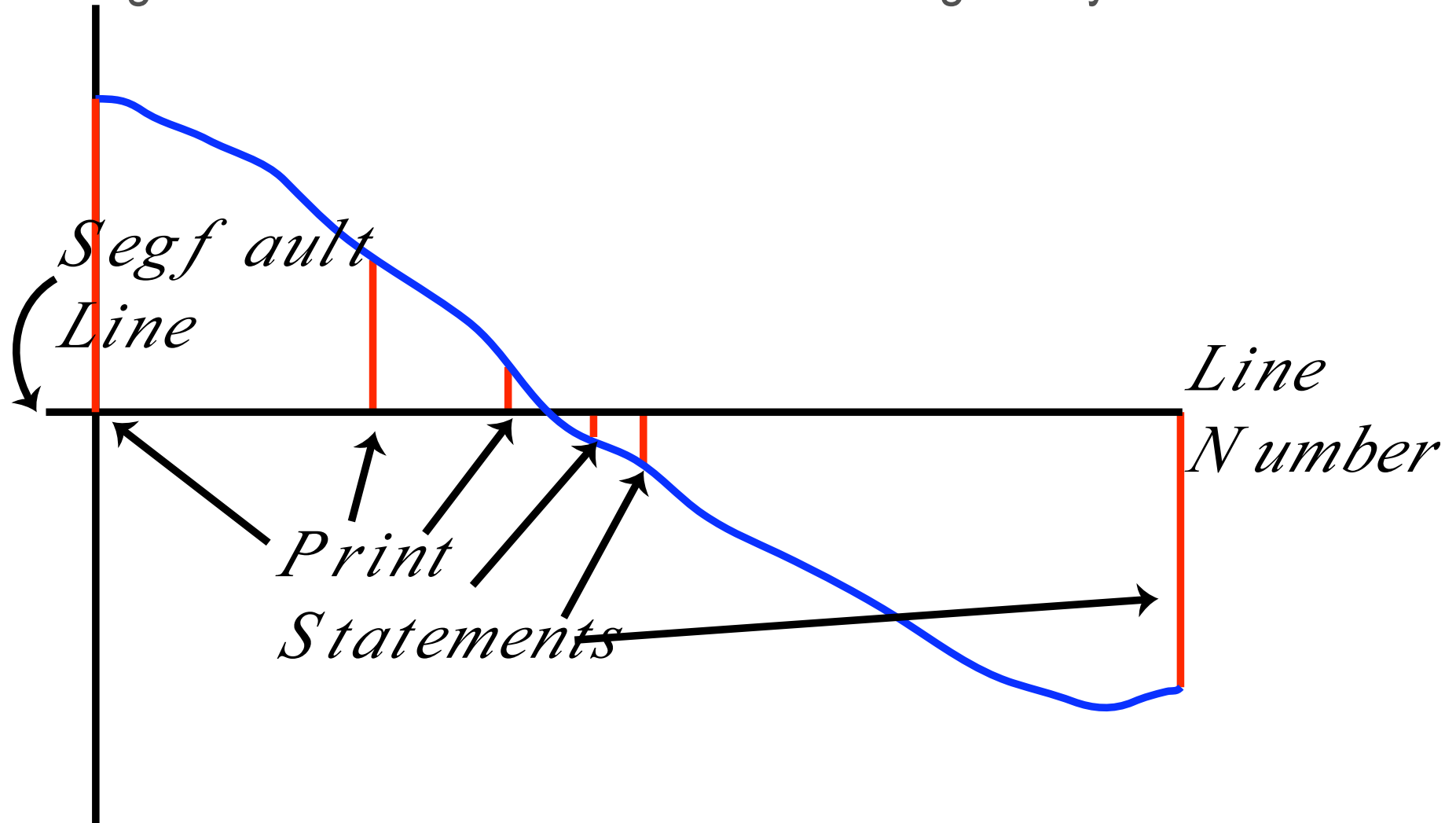
Print-Statement Debugging

- Analogous to bisection method of root finding – very slow!



Print-Statement Debugging

- Analogous to bisection method of root finding – very slow!



Code-Reading and Role-Playing



- Find a group of willing participants (alternatively, do it all yourself)
- Read through code from point of view of each processor at each step
- Create a great big chart that maps behavior of every process
- *Advantages:* helps you to learn code, and to learn who your true friends are ;)
- *Disadvantages:* time-consuming, tedious, not scalable (unless you are very popular)!

Arts & Crafts/Drawing



- Print out P copies of code
- Cut and paste relevant lines of code on individual papers for each processor
- On large paper or poster board, align papers at synchronization points, draw lines representing communication, etc.
- *Advantages:* get to play with scissors and glue, learn how code works
- *Disadvantages:* time-consuming, space-consuming, not scalable!

Using a Debugger



- Invoke executable within debugger
- Typically, must recompile with `-g` flag and optimizations off (for best fidelity)
- Advantages:
 - Debugger will concentrate on the state of the variables in the code, you figure out what it means
 - Time-saving: can often isolate problem in a single trial (especially segfaults)
- Disadvantages:
 - Some debuggers not available on all platforms
 - Sometimes code fails only with optimizations on, can be hard to locate exact place where things go wrong

Why Use Debuggers?

- Debuggers can save time
 - With print-statement debugging, must insert print statements into code, sift through print statements, and find error
 - Debugger allows you to find the line of code where problem occurs in a single trial (no bisection)
- Complexity of bugs grows with complexity of code
 - More lines of code, more potential for bugs
 - More complicated algorithm, more potential for errors
 - Parallelism only adds to complexity
 - Some bugs occur only at scale

What Can Debuggers Do for Me?

- Save time
- Allow user to concentrate on code, not background info
- View only variable values that are needed; view values not previously believed to be needed
- Pinpoint where things go wrong quickly
- Step through code and find cause of bug
- Run code at proper scale to find error

What Can Debuggers Do for Me?

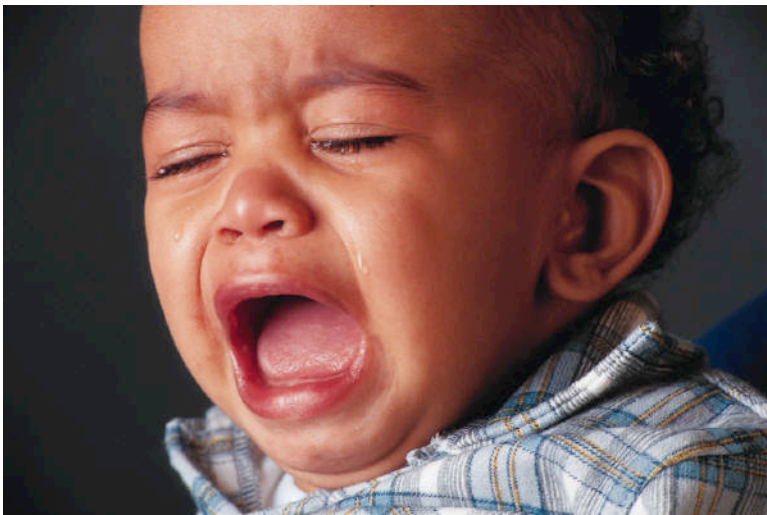
- Types of bugs
 - Segfaults
 - Memory errors
 - Algorithmic errors
 - Typos
 - “Improvements”
 - Things that happen only at scale
 - Etc.

A Come-to-Debuggers Moment

- There was once a grad student who could have been done with his/her dissertation SIX MONTHS EARLIER if he/she had been open to learning to use debuggers.
- “Oh no,” thought the grad student, “It will take me longer to learn to use a debugger than to just find this one last bug in my code.” But that was never the last bug. There was another, and another, and another...
- *It takes an initial investment to learn to use a debugger, but that investment will more than pay off in no time.*

A Come-to-Debuggers Moment

- That grad student can't have his/her 6 months back, but we can learn from the sad story and invest some time learning to use a debugger!



That unfortunate graduate student



You, having learned to use a debugger!